The Development of Software Certifier System (SoCfeS) : The Architecture and Design

Aziz Deraman
Vice Chancellor Office,
University Malaysia Terengganu, Kuala Terengganu, 21030, Terengganu, MALAYSIA
a.d@umt.edu.my

Jamaiah Haji Yahaya
College of Arts and Sciences, IT Building, Universiti Utara Malaysia, Sintok, 06010 Kedah, MALAYSIA
jamaiah@uum.edu.my

Abdul Razak Hamdan, Siti Zahidah Abdullah
Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia, ,Bangi, 43600, Selangor , MALAYSIA
{arh@ftsm.ukm.my, sza42@yahoo.com}

Abstract

*In this information age, most businesses are highly dependent on the availability of ICT services, especially on software application components. The interest on the acquisition of high quality software has increased among various stakeholders. However, some pertaining problems are still being debated such as: (i) defining mechanism for assessing software product quality; (ii) ensuring and offering software quality guarantee; and (iii) ensuring the continuous improvement of quality of software products. Therefore, a practical mechanism for assessment and certification is required to resolve these uncertainties. The fundamental model of certification or SCM-prod has been developed, evaluated and tested. It shows that the model and methodology are feasible and practical to be implemented in real world environment. Thus, a comprehensive model and support tool with intelligent aspects included is developed. The software named as **SoCfeS** (**So**ftware **C**ertifier **S**ystem) supports software certification process and continuous improvement intelligently.*

1. Introduction

In the new global economy and borderless world companies are competing to produce software which are claimed to be good and fulfill user's expectations and requirements. At the same time users complain that software are being delivered with bugs that need to be fixed and dissatisfied with the products [1][2]. Consistent with our observation in many local organizations and companies, we discover that software practitioners and software quality assurance (SQA) teams within the organizations could not guarantee the quality of the software either being developed in-house or purchased from external suppliers and vendors. The SQA teams agree that testing the software alone will not guarantee the quality status of the software. Testing the software alone will not guarantee that the software is good in quality. This leads to general perceptions among clients and practitioners that software industry as a whole lack of standard and mechanism for monitoring or assessing and ensuring software product quality.

2. Software certification model

Software certification is still a new concept in Malaysia but becoming increasingly popular in Europe and United States and at the same time many debates on this topic are reported. Previous survey conducted in Malaysia showed that even though it is a new idea and thought, it is still an acceptable concept perceived by respondents [3].

The term certification by definition is "an official document that says something is of good quality[4]. International Organisation for Standardisation defines certification as " a procedure by which a third party gives written assurance that a product, process or services conforms to specified

978-1-4244-4457-1/09/$25.00 ©2009 IEEE

characteristics" [5]. Certification is a possible approach to accomplish a continuous improvement and continuous assured quality of a software product. It is an alternative approach to attain quality continuously in its life span [6].

Issues in software quality have led to the proposal of software certification by independent, third party assessment. Involvement of an independent party in assessment of software may improve the quality of the assessment and thus guarantees and ensures the quality of the products [3]. Software assessment and certification can be viewed in three perspectives: process, product and people [7]. CMMI and PCMM are examples of software certification models by process and people approaches. Fauziah et al. developed a certification model by process approach named as SPAC that emphasized on five main entities: Development technology, project condition, people, environment, and process [8]. Each entity is broken down into sub entities and measures. Two assessment methods are constructed in this model, which by means of quality assessment and certification determination. At the end of the assessment exercise, the certification level is determined based on the quality assessment.

Software certification model by product quality approach is an alternative approach to assess software independent from development process. In this methodology, we assume that a good and systematic development process will not guarantee the good quality of software product. A western analogy says that "dirty water can run from a clean pipe" and it is true in software product development and construction. Therefore, the assessment and certification by product quality approach is relevant and needed. Several studies are conducted on this approach and among them are Voas [9], Morris et al. [10] and Heck & Eekelen [11]. Voas from Reliable Technology proposes a software certification method through involvement of end users. In this approach end users involve by delivering information regarding the usage of the software. One of the disadvantages of this approach is that users might overlook some of the important technical aspect of the software. User might not see or notice some of the technical requirement of the software during their assessment. Morris et al. develop a certification approach through developer's self-certification while LoQuso [11] invents the verification and validation technique in software certification model. LoQuso technique does not include the behavioural and human aspect of quality in the assessment. In our research we adopt a collaborative perspective approach and pragmatic quality model for assessment, which includes behavioural and human aspect of software quality. The certification and quality models proposed in this research will be discussed in the following section.

3. Our Previous Work: Software Product Certification Model (SCM-*Prod*)

The software certification by product quality approach is an acceptable approach of certifying software that operating in certain environment. A western analogy says that dirty water can run from clean pipes is believed to be true as a good software development processes do not guarantee the excellent quality of product. Thus, assessment of end product software must be independent from the development process. Previous studies [12,13] show that code analysis and testing software alone will not guarantee the quality of the product. Lauesen and Younessi (1998) conclude, "many defects cannot be found through analysis because they reflect tacit or undesirable requirements or can be observed only when the product is being used".

The software product certification model named SCM-*prod* is designed based on the following basis:

i) Assessment by independent body is an advantage to the user by conducting unbiased assessment. The independent certification is believed to be the only approach that user should trust and the demands for it are being heard from both publishers and users [17, 18]. While evaluation by the SQA team in the organization or the owner/users of the product is beneficial because they know well of the software and will reduce the time taken for assessment process. Thus in our approach the possibility of conducting assessment and certification of software product using collaborative perspective approach between the owner/users of the product, developers and independent assessor were being studied.

ii) The candidate software product is completed software and is operational in certain environment.

iii) The software quality factors apply in this research are derived from the ISO 9126 model with enhancement characteristics to accommodate other aspects of software quality requirements.

Figure 1 shows the components of SCM-*prod* model which consists of pragmatic quality factor (PQF), product certification repository, certification representation method, and assessment team. Refer to our previous publication for detail [14].

Pragmatic Quality Factor or PQF is the identified factor for quality assessment used in the certification process. PQF consists of two main components: the behavioural attributes and the impact attributes. The behavioural attributes deals with assessing software product to ensure the quality of the software and how it behaves in the environment. It includes efficiency, functionality, maintainability, portability, reliability, security and usability. While the impact attributes deal with how the software react and impact to the environment. These attributes include user perception and user requirement. These two components of quality produce a balance model between technical requirement and human factor.

The attributes defined in this model are the considered as the highest level in the hierarchy. These attributes then are broken down into several metrics and measures. The hierarchy model is adopted from IEEE software quality metrics framework [15]. The measures are the measurable quality aspects in this model and are based on perception scales obtained by the assessment team.

Another interesting feature of this model is the weighted scoring method (WSM) which applies different levels and categories of attribute with different weight factors. Literature suggests that each software quality attribute must not have the same level of importance in the real world environment to represent the actual business requirements. Survey conducted in this research indicated that there are some degrees of importance of each quality attribute and they can be classified into three layers namely low, moderate and high [16]. For each layer, a range of weight factor is assigned and recommended. The beauty of this approach is that software owner has a flexibility and authority to choose relevant weight values to reflect the organisation's and business's requirements and constraints. It is normal in business environment that in some situation certain quality attributes are more important than the others.

The model provides algorithms to measure software quality and software certification level based on identified standard. First algorithm is to measure quality status of each attribute based on the average score in assessment exercise. Second algorithm is to measure the certification level of the software product. Results from both algorithms are mapped into a certification representation model to determine the certification level (4, 3, 2 or 1) and its representation either of excellent, good, basic and acceptable, or poor.

The SCM-*prod* model provides procedures and guidelines for certifying software product operational in certain environment. Interviewee is defined in this model, which identify responsible person to evaluate items in the metrics. Thus, it gives fairer evaluation of the products because it names interviewee based on appropriateness and suitability of the person.

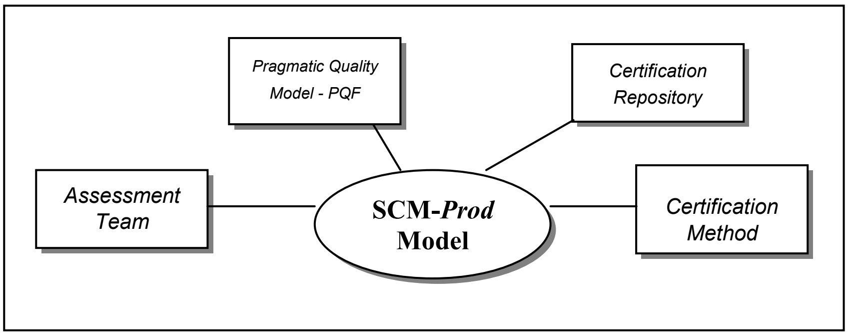

Figure 1. Components of SCM-*Prod* Model

4. SoCfeS : Software certifier system

The SCM-*prod* model was developed and implemented with minimal support by a toolset. We continue this research by enhancing the system that provides more support to the certification process. SoCfeS is an integrated software certifier system. This system is an embedded intelligent expert system to support the certification environment. The intelligent tool requires a self-learning capability with capturing knowledge from certification processes and experiences. Criteria of software assessment and certification might change and require additional new criteria to be included in future. Thus the intelligent toolset should capable to notice the changes and therefore will recommend to the environment of new modification assessment criteria. Figure 2 shows the meta model of SoCfeS.

The meta model explains the necessary components in the proposed system. The components are:-

- AQF – This represents the achieving quality factors for assessment. It is considered as the master file of quality factors. The notation used is *QF*.
- Method – Method represents methodology of the certification process. This applies the methodology and model (SCM-*prod*) discussed in the previous section. The notation used in *M*.
- SQF – It represent the selected quality factor. In this system, users have an opportunity to select their interested quality factors to be applied in the certification and assessment exercise depending on the organizations requirements. SQF $\in$ AQF.
- CKBase – The knowledgebase of certification. It captures and stores information of certification exercises in various software products.
- NQF – NQF represents the new quality factor identified in the environment. NQF is obtained by manipulation of experience and learning capabilities of the system supported by

Certification Knowledgebase (CKBase). The notation used is δ_{QF}.

- Certification Process – This represents the system that supports the certification process

The meta model above explains briefly on obtaining new quality factors that influence two components which by means of Method and AQF. Therefore, at different time the M and QF are formulated differently as the following :-

$$QF_{(t)} = QF_{(t)} + \delta_{QF}$$
$$M_{(t)} = M_{(t)} + \delta_{QF}$$

5. System Overview

The certification support tool developed in this research can serve in two different situations of handling certification exercise. First situation is implementing certification exercise through third party assessment or certification consultant. Second situation is implementing through self-certification by experts within the organisations. Therefore, our system design is focused to grip these two conditions and environments. With the later situation, the owner of the product may be able to assess their products at their own time interval so that the performance of their software products can be monitored and assessed.

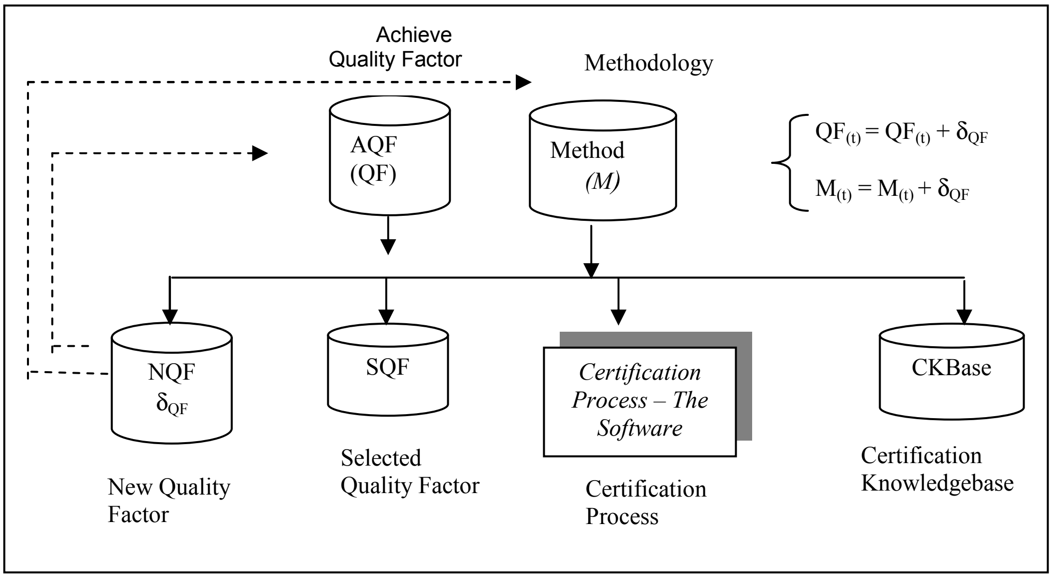

Figure 2. A Meta model of SoCfeS

The functional requirements for the support tool include the following: -

- Keep track of an identified software quality attributes and weights.
- Keep track of an identified set of quality metrics.
- Provide the templates of certifying analysis, results and reports.
- Provide the schema definition of proposed repository for future expansion and works.
- Provide flexibility in certification criteria.

Figure 3 is an architectural model, which illustrates the components of SoCfeS. High-level architectural model expressed as simple block diagrams where each sub-system is represented by a named rectangle, and lines indicate association between sub-systems. The architecture shows that SoCfeS system is connected to several modules: setup utility, product register, operation and services, maintenance, and learn & intelligent subsystems. It is also connected to certification databases and certification knowledgebase.

The setup utility is to prepare the process of certification that includes information on companies and organisations. The associated database setup is required to suit the individual organisation requirements and profile. The second module is the product register, which emphasises on setting up details of product and project. Product register includes tasks of criteria selection and weight assignment. Whilst the maintenance module is to provide facilities such as packaging and installation, certification tour, help menu and backup and restore utilities. The subsystem of operation is the main module of SoCfeS, which provides services of certification exercise. The tasks included are input data, data validation, certification process (that consists of two types of assessment: by individual attribute and product), report generation and expert review. The learn and intelligent module is an intelligent subsystem that offers recommendation to the user of new quality attributes based on information from the knowledge base.

The certification database consists of twenty individual tables that capture the necessary data in this system. The 20 individual tables are constructed to store data on quality; weight values and related supporting data. The data collected is then being analyzed to form a certification knowledgebase for further analysis by the system. The detail of the intelligent system in not covered in this paper.

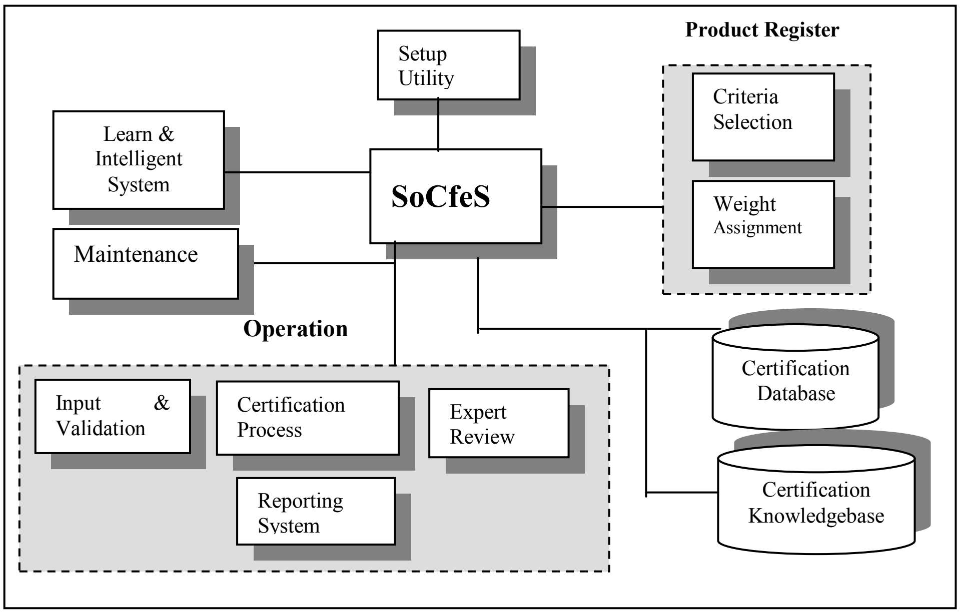

Figure 3. The general architecture of SoCfeS System

6. The Implementation

For the first phase implementation, SoCfeS system is developed using Visual C# programming tool and MS-Access for database management system. MS-Access is used since its tables are not so large and can easily manipulate from C# through ADO connection.

Visual C# is an event-driven, visual programming language in which programs are created using an Integrated Development Environment (IDE). With the IDE, a C# program can be created, ran, tested and debugged conveniently, thereby reducing the time it takes to produce a working program. At least nine classes were built in this project. These classes are: - *Cert_main.cs, ClassEff.cs, ClassFunct.cs, ClassMaint.cs, ClassInteg.cs, ClassReli.cs, ClassPort.cs, ClassUsab.cs* and *ClassUser.cs.* The method connects with table *Certrifiction.mdb* that contains all the relevant tables. SQL commands are used to retrieve and manipulate data from tables.

SoCfeS contains several window forms in handling graphical user interface (GUI). It allows users to interact visually with a program. This project contains at least eleven forms and each form handles different circumstances. The forms are: *Form_InputEfficiency, Form_InputFunct, Form_InputInteg, Form_InputMaint, Form_InputPort, Form_InputReli, Form_InputUsab, Form_InputUserF, Form1, Form_Calculate, Form_Weight.*

The implementation has shown that the use of software certification model for assessing and certifying software product is viable. SoCfeS, the toolset is developed in-house and currently in testing phase. Figure 4 and 5 illustrate a few screen snapshots of SoCfeS. The next stage is to test the toolset in certification exercise collaboration with software industry. The completed and detail of the system design and implementation will be published in the near future.

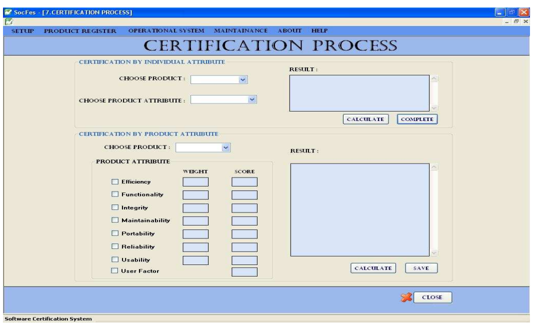

Figure 4. Screen snapshot of SoCfeS (certification process)

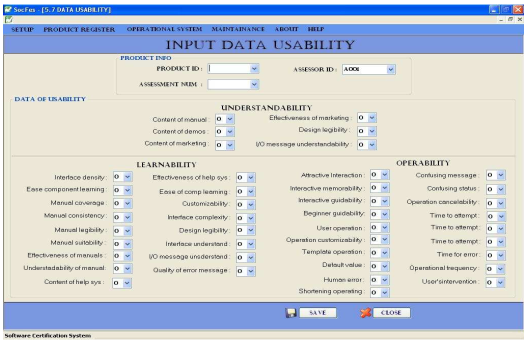

Figure 5. Screen snapshot of SoCfeS (input data - usability)

7. Conclusion

A model that may be used to certify software product has been presented in this paper. This model has been developed in a goal-directed way in order to meet the needs of the different interest groups associated with software quality. The model is a practical model of certification, which was evaluated and tested, in real case studies in Malaysia. We extend this model to produce an integrated model to meet wider requirements in certification process. SoCfeS is an intelligent software certifier system developed in this research. This paper explained the extension model and discussed the architecture of SoCfeS that consists of several modules of certification processes, an intelligent module and expert system embedded for supporting the continuous improvement in certification environment. It supports self-learning capability with knowledge over the certification environment. In future SoCfeS updates the quality attributes and certification component based on knowledge captured through certification data and exercises. This is important as the quality attributes might change over time based on current requirements and specification.

8. Acknowledgement

This project is funded by Malaysian Ministry of Science, Technology and Innovation.

9. References

[1] Whittaker, J. A., & Voas, J. M., 50 years of software: Key principles for quality. *IEEE IT Pro* (Nov/Dec), 2002, pp. 28-35.

[2] Denning, P.J. What is software quality? *A Commentary from Communications of ACM* (January), 1999.

[3] Yahaya, J.H., Deraman, A. & Hamdan, A.R., Software quality and certification: Perception and practices in Malaysia. *Journal of ICT (JICT)*, **5**(Dec), 2006, pp. 63-82.

[4] Longman. 2003. *Dictionary of Contemporary English*. Essex: Pearson Education Limited.

[5] Rae, A. K., Hausen, H.L. & Robert, P., *Software Evaluation for Certification : Principles, Practice and Legal Liability.* Middlesex, UK: McGraw-Hill, 1995.

[6] Deraman, A., Yahaya, J.H., Baharom, F., Fadzlah, A.F.A. & Hamdan A.R. Continuous quality improvement in software certification environment. Proceedings of the International Conference on Electrical Engineering and Informatics (ICEEI 2007), 2007, pp.11-17.

[7] Voas, J., The software quality certification triangle: Crosstalk. *The Journal of Defense Software Engineering* (November), 1998, pp. 12-14.

[8] Fauziah Baharom, Aziz Deraman, Abdul Razak Hamdan & Jamaiah H. Yahaya. 2008. Software Certification towards Assuring and Improving the Quality of Software Development Process. *Proceedings of Malaysian Software Engineering Conference (mySec'08), Kuala Terengganu,* 16-17 Dec 2008.

[9] Voas, J., User participation-based software certification. Proceedings of Eurovav 1999*,* pp. 267-276.

[10] Morris, J., Lee, G., Parker, K., Bundell, G.A., & Lam, C.P., Software component certification. *IEEE Computer* (September), 2001, pp. 30-36.

[11] Heck, P & Eekelen, M. v. "LaQuSo Software Product Certification Model (LSPCM)" *http://alexandria.tue.nl/repository/books/633706.pdf 2008.* [16 July 2008].

[12] S. Lauesen, & H. Younessi, "Is Software Quality Visible in the Code?", *IEEE Software,* July/August 1998, pp. 69-73,.

[13] G.R. Dromey, "Software Quality - Prevention versus Cure?", *Software Quality Journal*, 2003, 11, pp. 197-210.

[14] Yahaya, J.H., Deraman, A. & Hamdan, A.R., 2008. Software Certification Model Based on Product Quality Approach. *Journal of Sustainability Science and Management,* Vol. 3 No 2, December 2008, pp. 14-29.

[15] IEEE: IEEE standard for a software quality Metrics Methodology, http://ieeexplore.ieee.org/xpl/standards.jsp , 1993.

[16] Yahaya, J.H., Deraman, A. & Hamdan, A.R., Software product certification model: Classification of quality attributes. *The First Regional Conference of Computational Science and Technology (RCCST 07), Kota Kinabalu,* 2007, pp. 436-440.

[17] M.F. Bertoa, J.M. Troya, and A. Vallecillo, "A Survey on the Quality Information Provided by Software Component Vendors", *in Proceedings of 7th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2003),*2003, pp. 25-30.

[18] J. Voas, "Certification: Reducing the hidden cost of poor quality", *IEEE Software*, July/August 1999, pp. 22-25.